



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:  
**26.06.2002 Bulletin 2002/26**

(51) Int Cl.7: **H04L 29/06, H04L 12/56**

(21) Application number: **01309084.0**

(22) Date of filing: **25.10.2001**

(84) Designated Contracting States:  
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU**  
**MC NL PT SE TR**  
Designated Extension States:  
**AL LT LV MK RO SI**

(72) Inventor: **Hartop, Scott**  
**London NW11 0DG (GB)**

(74) Representative: **Williamson, Brian et al**  
**International IP Department,**  
**NCR Limited,**  
**206 Marylebone Road**  
**London NW1 6LY (GB)**

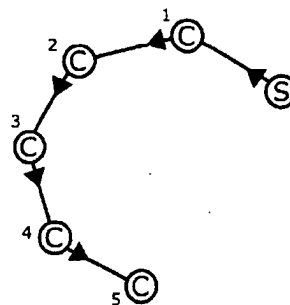
(30) Priority: **20.12.2000 GB 0031157**

(71) Applicant: **NCR International, Inc.**  
**Dayton, Ohio 45479 (US)**

(54) **Streaming of data in a peer-to-peer architecture**

(57) A method of optimising data streaming in a peer-to-peer architecture that comprises a plurality of clients in a chain, a peer-to-peer data streaming system having such architecture, and a client terminal for use in that system. Each client, except the last client in the chain, monitors its own bandwidth, informs a succeeding client in the chain of that bandwidth, compares its own bandwidth with the bandwidth of a preceding client in the chain and, in response to a difference between the compared bandwidths, reorders its position among the clients in the chain.

The chain thus dynamically reorganises itself to stream data more efficiently and with higher, more reliable throughput, reducing the processing power necessary to stream the data and enabling higher quality to be achieved compared to the existing essentially client/server internet infrastructure. This approach also solves the 'bottle-neck' problem within the cascaded streaming path by continuously organising the participating terminals into the most efficient configuration, without interrupting the streamed data.



**Figure 2**

## Description

[0001] This invention relates to streaming of data, and more generally to the distribution of rich media and data on-line and over any network, notably the Internet.

[0002] Streaming technology is a major growth area in the Internet field. It helps to satisfy public demand for large multimedia files by allowing parts of such a file to be displayed, played or otherwise accessed and used while other parts of the file are still downloading. In this way, streaming helps users whose terminals have insufficient access speed, memory and processing capabilities for them to be able to download complete multimedia files quickly enough to enjoy their use.

[0003] Examples of streaming formats are RealVideo and RealAudio developed by RealNetworks, Inc. (all trade marks acknowledged). With suitable plug-ins on their browser programs, users with even modestly-specified terminals and modems can enjoy streamed data such as a live audio/video feed from a concert.

[0004] Streaming relies upon the client terminal collecting data and sending that data as a steady stream to an application that processes the data, for example by converting that data to sound and/or pictures. Despite the use of a buffer to store an excess of incoming data and hence to insulate the application from interruptions in the incoming data stream, it often happens that the data does not arrive quickly enough to replenish the buffer. For example, network congestion can arise due to the essentially client/server architecture of the Internet. The result is a lack of smoothness in the data stream presented to the application and consequential loss in the quality of the user's experience, manifested by interruptions and other degradation.

[0005] Even if the data stream is smooth enough to allow continuity, the quality of the data, for example in terms of image resolution and image size, is often poor. Put simply, the quality of the data equates to the rate at which information is transmitted, in terms of the amount of data transmitted in a given time, and this rate is compromised in an effort to ease downloading, to make it easier to keep the buffer replenished, and to minimise interruptions in the data stream.

[0006] In today's predominantly client/server Internet architecture, high concurrency rates place bandwidth-intensive demands on servers dedicated to the streaming of continuous data. This drastically compromises the quality and reliability of the end-user's experience. It also results in high hardware overheads for streaming service providers, who need to maintain banks of specialised servers.

[0007] Against this background, the invention resides in a method of optimising data streaming in a peer-to-peer architecture comprising a plurality of clients in a chain, the method comprising each client monitoring its own bandwidth, informing a succeeding client in the chain of that bandwidth, comparing its own bandwidth with the bandwidth of a preceding client in the chain and,

in response to a difference between the compared bandwidths, reordering its position among the clients in the chain.

[0008] Similarly, the invention can be expressed in terms of a peer-to-peer data streaming system comprising a plurality of clients in a chain, each client including bandwidth-monitoring means for monitoring its own bandwidth, communication means for informing a succeeding client in the chain of that bandwidth, comparison means for comparing its own bandwidth with the bandwidth of a preceding client in the chain, and reconfiguration means responsive to a difference between the compared bandwidths to reorder its position among the clients in the chain.

[0009] From the client perspective, the invention resides in a client terminal for use in a peer-to-peer data streaming system that comprises a plurality of client terminals in a chain, the client terminal being configured or programmed to include bandwidth-monitoring means for monitoring its own bandwidth, communication means for informing a succeeding client terminal in the chain of that bandwidth, comparison means for comparing its own bandwidth with the bandwidth of a preceding client terminal in the chain, and reconfiguration means responsive to a difference between the compared bandwidths to reorder its position among the client terminals in the chain.

[0010] The client terminal can of course be a client computer such as a home or office PC but can take other forms such as, without limitation, a set-top box, a games console, a networked hi-fi system or other network device.

[0011] The invention therefore enables and provides a streaming architecture which uses a piece of software to link any networked computing device (referred to herein as clients or peers) in a continuous, dynamically self-organising peer-to-peer chain for the purpose of streaming any kind of data more efficiently and with higher, more reliable throughput. That software is optionally downloadable or may be distributed in any convenient format. The invention also includes a client/server co-ordinating element to originate and monitor the chain, capable of handling appropriate content management and permissions functions.

[0012] This invention transfers streaming from the known client/server architecture to a peer-to-peer architecture, exponentially reducing the processing power necessary to stream any continuous data and enabling far higher quality (i.e. a higher rate of information transfer, with less interruptions) to be achieved within the existing internet infrastructure.

[0013] This invention also solves the potential 'bottleneck' effect within the cascaded streaming path by dynamically self-organising the participating terminals or networked devices into the most efficient configuration at any given moment, without interrupting the streamed information.

[0014] In the system or terminal of the invention, a cli-

ent or client terminal preferably includes address-providing means for receiving and storing the address of a preceding or succeeding client or client terminal in the chain and providing that address to, respectively, the succeeding or preceding client or client terminal in the chain. Thus, for example, each client can identify a preceding client in the chain to the succeeding client in the chain. Similarly, the detecting client can identify a succeeding client in the chain to the preceding client in the chain.

**[0015]** If a detecting client detects that its bandwidth is greater than that of the preceding client in the chain then, in response, it opens a connection with a client upstream of the preceding client. In parallel, the preceding client in the chain can open a connection with the identified succeeding client.

**[0016]** In apparatus terms, the comparison means of a client or client terminal is associated with connection means for receiving the address of, and opening a connection with, a client or client terminal upstream of the preceding client or client terminal if the comparison means detects that the bandwidth of its associated client or client terminal is greater than that of the preceding client or client terminal in the chain.

**[0017]** The or each of the new connections is preferably opened concurrently with pre-existing connections between clients in the chain. Once the or each concurrent connection has been made to a client, the or each associated pre-existing connection to that client can be dropped. In that event, the client advantageously switches to reading local buffer memory before the pre-existing connection is dropped.

**[0018]** In the reordered chain, therefore, the detecting client can receive streamed data from the client upstream of the preceding client and can forward that streamed data to the preceding client. For example, the pre-existing connection between the preceding client and the detecting client can be reversed, or a replacement connection can be opened between the preceding client and the detecting client. Similarly, in the reordered chain, the succeeding client can receive streamed data from the preceding client.

**[0019]** Once the chain has been reordered, a client preferably synchronises a timecode of data in local buffer memory with a timecode of data received from a new streamed data input source before switching to data received from that source. The client can then replenish its local buffer memory.

**[0020]** The invention extends to an optionally-downloadable software application that is adapted to configure or program a client terminal to implement the inventive features expressed above.

**[0021]** In order that this invention may be more readily understood, reference will now be made, by way of example, to the accompanying drawings in which:

Figure 1 is a diagram of a first step in the process of the invention, in which communication is estab-

lished between a new client and a co-ordinating server;

Figure 2 is a diagram of a second step in the process of the invention, in which communication is established between the new client and another client from which the new client receives streamed data and forwards that data to other clients along a peer-to-peer chain;

Figure 3 is a diagram of a third step in the process of the invention, in which a further client joins the chain of Figure 2 and concurrent connections are made between clients in preparation for re-routing data flow among the clients in the chain; and

Figure 4 is a diagram of a fourth step in the process of the invention, in which data flow has been re-routed among the clients in the chain.

**[0022]** A consumer interested in receiving any streamed data (for example, a live audio/video feed from a concert) first of all subscribes to the service. Having been granted membership, that consumer is able to download or otherwise obtain a software application to the networked device or terminal that the consumer intends to use to interpret or decode the streamed information. This application could be branded, can be customisable (for example by means of 'skins' that impose various attributes of appearance) and may feature additional functionality, but its core jobs are as follows:

1. Communicating briefly with a co-ordinating server to be allocated a starting place in a peer-to-peer chain of networked devices.

2. Having dropped this server connection, establishing a connection to the IP address allocated by the server, from which address the desired streaming signal can be received.

3. Providing 'repeater' functionality such that the application can pass on the encoded signal to a succeeding terminal or networked device in the chain (being the subsequent node in the peer-to-peer connection) without interrupting the signal, and also simultaneously decoding the incoming signal and relaying this to the appropriate playback device or application associated with the user's networked device. Of course, depending on the playback device or application, interpreting the codec may not be necessary.

4. As background functions:

- (a) obtaining bandwidth information by monitoring the connection speed of the host networked device or terminal, for example by averaging

over a period of time a series of regular and frequent counts of the rate at which data is received (suitably measured in bits per second, say every 500 milliseconds);

(b) providing a continuous (or frequently enough to be effectively continuous) update of this bandwidth information to the succeeding networked device or terminal in the chain, together with the IP address of the preceding networked device or terminal in the chain for use when adjusting the client streaming order, as will be described; and

(c) comparing its own throughput rate with that of the preceding networked device or terminal in the chain.

5. As a result of learning that its host networked device or terminal is operating faster than the preceding networked device or terminal in the chain from which data is currently being received, the application manages the process of creating a new connection with the next preceding networked device or terminal, one link further up the chain. This involves the application seamlessly adjusting the position of its associated networked device in the chain or streaming cascade without disturbing the relative position of the streamed information, and then terminating the previous 'bottle-necking' connection.

6. Log the streaming activity of the machine for corroboration with the co-ordinating server at an appropriate future time.

[0023] Referring firstly then to Figure 1 of the drawings, a new client C contacts a co-ordinating server S in normal client/server architecture. The server S processes the request according to subscription/membership details uploaded from the client C and once access is confirmed, passes to the client C a decryption key for the requested stream and the IP address of the last client that the server S processed and added to a chain.

[0024] Figure 2 shows the next step in the process, in which the new client, which will henceforth be called Client 5, drops the connection to the server and switches to the allocated client identified by the server (this being Client 4, formerly the last in the chain). Client 5 then begins to receive streamed data from Client 4. That data is time-coded frame-by-frame.

[0025] In Figure 3, another client, namely Client 6, has joined the chain of Figure 2. The abovementioned bandwidth monitoring function run by Client 5 and taking bandwidth information from Client 4 detects that Client 5 is faster than Client 4, Client 4 thus presenting a potential bottleneck in comparison with Client 5. To avoid this, Client 5 briefly sends the IP address of Client 6 to

Client 4, whereupon Client 4 forges a concurrent connection with Client 6. Meanwhile, Client 5 forges a concurrent connection with Client 3, whose IP address is already known to Client 5 because Client 4 included that IP address with the bandwidth information it sent to Client 5. That done, and in preparation for the steps shown in Figure 4, Clients 4, 5 and 6 all switch to reading their local buffer memory of the streamed signal to preserve continuity in their output to their users.

[0026] In Figure 4, having adjusted buffer times and synchronised timecode where necessary to ensure a contiguous stream:

Client 5 switches to receiving and processing the signal from Client 3 and reverses the direction of its connection to Client 4. In other words, Client 5 sends its cascaded output to the IP address of Client 4 (a new replacement connection may need to be established here to achieve this);

Client 4 switches to receiving and processing the streamed signal from Client 5 and outputting to Client 6; and

the concurrent (but now unused) connections between Clients 3 and 4 and Clients 5 and 6 are dropped.

[0027] Using the timecode of the current frame being played from buffer memory to synchronise with the timecode of the new streamed input source, all clients revert to processing the streamed information and replenish their local buffers as quickly as connection speed allows.

[0028] It will be apparent from the foregoing that the chain has optimised itself according to local rules, without server co-ordination. Changes in available bandwidth at any point in the chain are dynamically resolved as they occur.

[0029] The last client in the chain can never be replaced via this self-organising process in order that the co-ordinating server handling requests to join the chain knows the IP address that defines where to instruct newcomers to connect, bearing in mind that the server is not connected to a client in the chain while the chain reconfigures itself during self-optimisation.

[0030] By means of the invention, a domestic user or consumer can, for example, launch the software, join a peer-to-peer chain streaming a piece of live theatre which is already in progress and automatically enjoy optimum network performance, regardless of the physical ceiling of the connection, whilst monitoring the show on their preferred home entertainment device such as a PC, a home cinema system and so on. The user may already have subscribed to and paid for this content, or could be billed on quitting the chain based on the amount of content actually streamed by their terminal.

## Claims

1. A method of optimising data streaming in a peer-to-peer architecture comprising a plurality of clients in a chain, the method comprising each client (5) monitoring its own bandwidth, informing a succeeding client (6) in the chain of that bandwidth, comparing its own bandwidth with the bandwidth of a preceding client (4) in the chain and, in response to a difference between the compared bandwidths, reordering its position among the clients in the chain. 5
2. The method of Claim 1, wherein each client (5) identifies a preceding client (4) in the chain to the succeeding client (6) in the chain. 10
3. The method of Claim 1 or Claim 2, wherein a detecting client (5) detects that its bandwidth is greater than that of the preceding client (4) in the chain and, in response, opens a connection with a client (3) upstream of the preceding client (4). 15
4. The method of Claim 3, wherein the detecting client (5) identifies a succeeding client (6) in the chain to the preceding client (4) in the chain. 20
5. The method of Claim 4, wherein the preceding client (4) opens a connection with the identified succeeding client (6). 25
6. The method of any of Claims 3 to 5, wherein the or each of said connections is opened concurrently with pre-existing connections between clients in the chain. 30
7. The method of any of Claims 3 to 6, wherein after the or each concurrent connection has been made to a client, the or each associated pre-existing connection to that client is dropped. 35
8. The method of Claim 7, wherein the client switches to reading local buffer memory before the pre-existing connection is dropped. 40
9. The method of any of Claims 3 to 8, wherein, in the reordered chain, the detecting client (5) receives streamed data via said connection from the client (3) that was upstream of the preceding client (4). 45
10. The method of Claim 9, wherein the detecting client (5) sends streamed data to the preceding client (4). 50
11. The method of Claim 10, wherein the pre-existing connection between the preceding client (4) and the detecting client (5) is reversed. 55
12. The method of Claim 11, wherein a replacement connection is opened between the preceding client (4) and the detecting client (5).
13. The method of any of Claims 9 to 12 when appendant to Claim 5, wherein, in the reordered chain, the succeeding client (6) receives streamed data via said connection from the preceding client (4).
14. The method of any preceding Claim, wherein after the chain has been reordered, a client synchronises a timecode of data in local buffer memory with a timecode of data received from a new streamed data input source before switching to data received from that source.
15. The method of any preceding Claim, wherein a client replenishes its local buffer memory after the chain has been reordered.
16. A peer-to-peer data streaming system comprising a plurality of clients in a chain, each client (5) including bandwidth-monitoring means for monitoring its own bandwidth, communication means for informing a succeeding client (6) in the chain of that bandwidth, comparison means for comparing its own bandwidth with the bandwidth of a preceding client (4) in the chain, and reconfiguration means responsive to a difference between the compared bandwidths to reorder its position among the clients in the chain.
17. A client terminal (5) for use in a peer-to-peer data streaming system that comprises a plurality of client terminals in a chain, the client terminal (5) being configured or programmed to include bandwidth-monitoring means for monitoring its own bandwidth, communication means for informing a succeeding client terminal (6) in the chain of that bandwidth, comparison means for comparing its own bandwidth with the bandwidth of a preceding client terminal (4) in the chain, and reconfiguration means responsive to a difference between the compared bandwidths to reorder its position among the client terminals in the chain.
18. The system of Claim 16 or the terminal of Claim 17, wherein a client or client terminal (5) includes address-providing means for receiving and storing the address of a preceding (4) or succeeding (6) client or client terminal in the chain and providing that address to, respectively, the succeeding (6) or preceding (4) client or client terminal in the chain.
19. The system or terminal of any of Claims 16 to 18, wherein the comparison means of a client or client terminal (5) is associated with connection means for receiving the address of, and opening a connection with, a client or client terminal (3) upstream of the preceding client or client terminal (4) if the compar-

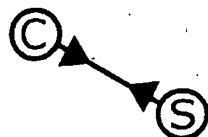
ison means detects that the bandwidth of its associated client or client terminal (5) is greater than that of the preceding client or client terminal (4) in the chain.

- 5
20. The system or terminal of Claim 19, wherein the connection means is capable of opening a connection concurrently with a pre-existing connection between clients or client terminals in the chain.
- 10
21. The system or terminal of Claim 20, wherein the connection means is responsive to making the concurrent connection to drop the associated pre-existing connection.
- 15
22. The system or terminal of Claim 21, wherein the connection means is associated with switch means for switching the client or client terminal to read local buffer memory before the pre-existing connection is dropped.
- 20
23. The system or terminal of any of Claims 19 to 22, wherein the connection means is capable of reversing a pre-existing connection between clients or client terminals in the chain.
- 25
24. The system or terminal of any of Claims 16 to 23, wherein a client or client terminal (5) comprises data synchronising means for synchronising a timecode of data in local buffer memory with a timecode of data received from a new streamed data input source.
- 30
25. The system or terminal of Claim 24, wherein a client or client terminal (5) comprises switch means responsive to the data synchronising means to switch to data received from the new streamed data input source when said timecodes are synchronised.
- 35
26. An application adapted to configure or program a client terminal (5) in accordance with any of Claims 17 to 25.
- 40

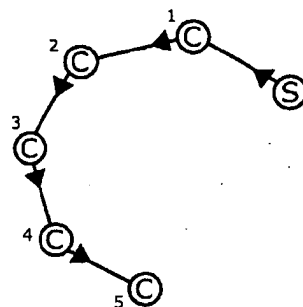
45

50

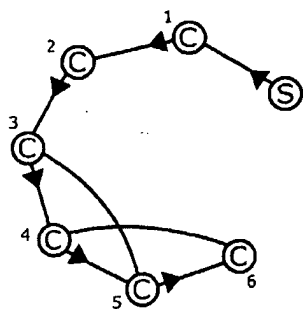
55



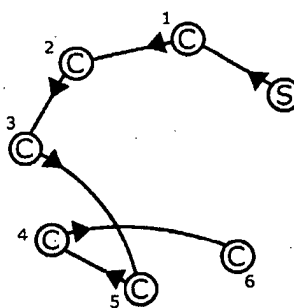
**Figure 1**



**Figure 2**



**Figure 3**



**Figure 4**



European Patent  
Office

# EUROPEAN SEARCH REPORT

Application Number  
EP 01 30 9084

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	US 5 640 384 A (DU YONGGANG) 17 June 1997 (1997-06-17) * abstract * * column 1, line 24-35 * * column 2, line 26-37 * * figures 2A-2C *	1-26	H04L29/06 H04L12/56
A	US 5 485 455 A (ANDLAUER PHIL ET AL) 16 January 1996 (1996-01-16) * abstract * * column 1, line 56-62 * * column 4, line 21-25, 39-41 * * figure 13 *	1-26	
A	WO 00 02345 A (GILBERT MARTYN) 13 January 2000 (2000-01-13) * abstract * * page 4, line 24 - page 5, line 7 * * figure 1 *	1-26	
A	JONAS K ET AL: "Audio streaming on the Internet. Experiences with real-time streaming of audio streams" INDUSTRIAL ELECTRONICS, 1997. ISIE '97., PROCEEDINGS OF THE IEEE INTERNATIONAL SYMPOSIUM ON GUIMARAES, PORTUGAL 7-11 JULY 1997, NEW YORK, NY, USA, IEEE, US, 7 July 1997 (1997-07-07), pages SS71-SS76, XP010265142 ISBN: 0-7803-3936-3 * the whole document *	1-26	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L
The present search report has been drawn up for all claims			
Place of search MUNICH		Date of completion of the search 23 April 2002	Examiner Gabriel, C
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1503 (03.02) (P04001)



**ANNEX TO THE EUROPEAN SEARCH REPORT  
ON EUROPEAN PATENT APPLICATION NO.**

EP 01 30 9084

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.  
The members are as contained in the European Patent Office EDP file on  
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

23-04-2002

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5640384	A	17-06-1997	EP 0717900 A1	26-06-1996
			WO 9535612 A2	28-12-1995
			JP 9503363 T	31-03-1997
US 5485455	A	16-01-1996	AT 205652 T	15-09-2001
			AU 678687 B2	05-06-1997
			AU 1607895 A	15-08-1995
			AU 681062 B2	14-08-1997
			AU 6190296 A	03-10-1996
			AU 681063 B2	14-08-1997
			AU 6190396 A	03-10-1996
			DE 69522666 D1	18-10-2001
			EP 0741937 A1	13-11-1996
			EP 0746175 A2	04-12-1996
			EP 0746176 A2	04-12-1996
			JP 9508509 T	26-08-1997
			WO 9520850 A1	03-08-1995
			US 5491694 A	13-02-1996
			US 5521910 A	28-05-1996
			US 5790546 A	04-08-1998
WO 0002345	A	13-01-2000	AU 4634599 A	24-01-2000
			CN 1312997 T	12-09-2001
			EP 1142259 A2	10-10-2001
			WO 0002345 A2	13-01-2000
			NO 20006622 A	01-03-2001

EPO FORM P439

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82